Solution 1: L0 Regularization

Consider the regression learning setting, i.e., $\mathcal{Y} = \mathbb{R}$, and feature space $\mathcal{X} = \mathbb{R}^p$. Let the hypothesis space be the linear models:

$$\mathcal{H} = \{f(\mathbf{x}) = \boldsymbol{\theta}^\top \mathbf{x} \mid \boldsymbol{\theta} \in \mathbb{R}^p\}.$$

Suppose your loss function of interest is the L2 loss $L(y, f(\mathbf{x})) = \frac{1}{2}\big(y - f(\mathbf{x})\big)^2$. Consider the L_0-regularized empirical risk of a model $f(\mathbf{x} \mid \boldsymbol{\theta})$:

$$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_0 = \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right)^2 + \lambda \sum_{i=1}^{p} \mathbb{1}_{|\theta_i| \neq 0}.$$

Assume that $\mathbf{X}^T\mathbf{X} = \boldsymbol{I}$, which holds if $\mathbf{X}$ has orthonormal columns. Show that the minimizer $\hat{\theta}_{\text{L0}} = (\hat{\theta}_{\text{L0},1}, \ldots, \hat{\theta}_{\text{L0},p})^\top$ is given by

$$\hat{\theta}_{\text{L0},i} = \hat{\theta}_i \mathbb{1}_{|\hat{\theta}_i| > \sqrt{2\lambda}}, \quad i = 1, \ldots, p,$$

where $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_p)^\top = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ is the minimizer of the unregularized empirical risk (w.r.t. the L2 loss).

For this purpose, use the following steps:

(i) Derive that

$$\arg\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{p} -\hat{\theta}_i\theta_i + \frac{\theta_i^2}{2} + \lambda \mathbb{1}_{|\theta_i| \neq 0}.$$

(ii) Note that the minimization problem on the right-hand side of (i) can be written as $\sum_{i=1}^{p} g_i(\theta_i)$, where

$$g_i(\theta) = -\hat{\theta}_i\theta + \frac{\theta^2}{2} + \lambda \mathbb{1}_{|\theta| \neq 0}.$$

What is the advantage of this representation if we seek to find the $\boldsymbol{\theta}$ with entries $\theta_1, \ldots, \theta_p$ minimizing $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$?

(iii) Consider first the case that $|\hat{\theta}_i| > \sqrt{2\lambda}$ and infer that for the minimizer θ_i^* of g_i it must hold that $\theta_i^* = \hat{\theta}_i$.

Hint: Show that $g_i(\hat{\theta}_i) < 0 = g_i(0)$ and argue that the minimizer must have the same sign as $\hat{\theta}_i$.

(iv) Derive that $\theta_i^* = \hat{\theta}_i \mathbb{1}_{|\hat{\theta}_i| > \sqrt{2\lambda}}$, by using (iii) (and also still considering the case $|\hat{\theta}_i| > \sqrt{2\lambda}$).

(v) Consider the complementary case of (iii) and (iv), i.e., $|\hat{\theta}_i| \leq \sqrt{2\lambda}$, and infer that for the minimizer θ_i^* of g_i it must hold that $\theta_i^* = 0$.

Hint: What is $g_i(0)$? Consider $\tilde{g}_i(\theta) = -\hat{\theta}_i\theta + \frac{\theta^2}{2} + \lambda$ which is the smooth extension of g_i. What is the relationship between the minimizer of g_i and the minimizer of $\tilde{g}_i$?

Solution 2: Regularization

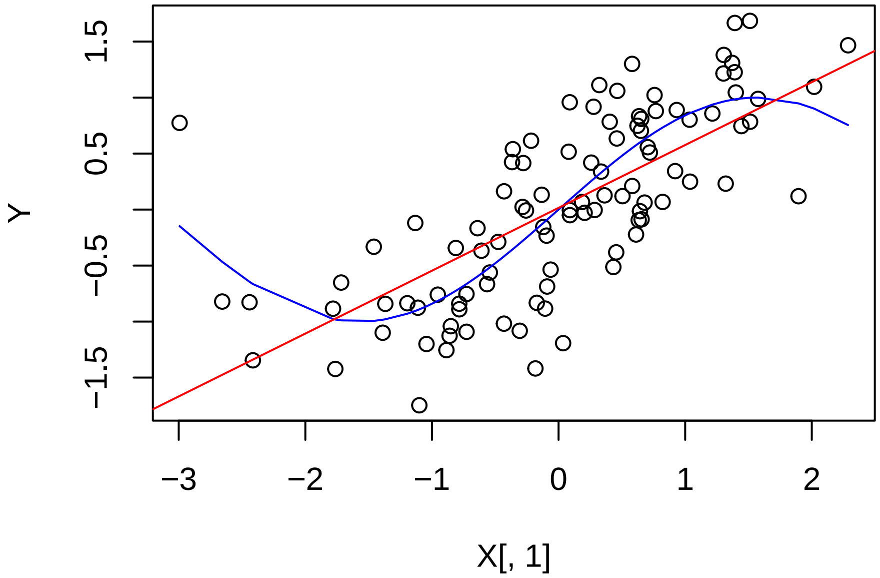
(a)
```r
set.seed(42)
n = 100
p_add = 100
# create matrix of features
X = matrix(rnorm(n * (p_add + 1)), ncol = p_add + 1)

Y = sin(X[,1]) + rnorm(n, sd = 0.5)
```

(b) Demonstration of

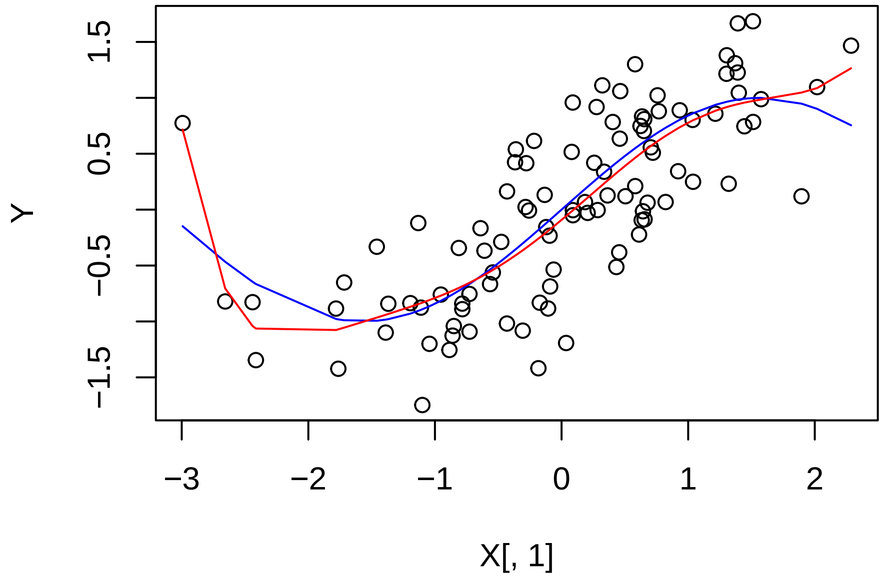
- underfitting:

```
plot(X[,1], Y)
points(sort(X[,1]), sin(sort(X[,1])), type="l", col="blue")
abline(coef(lm(Y ~ X[,1])), col="red")
```



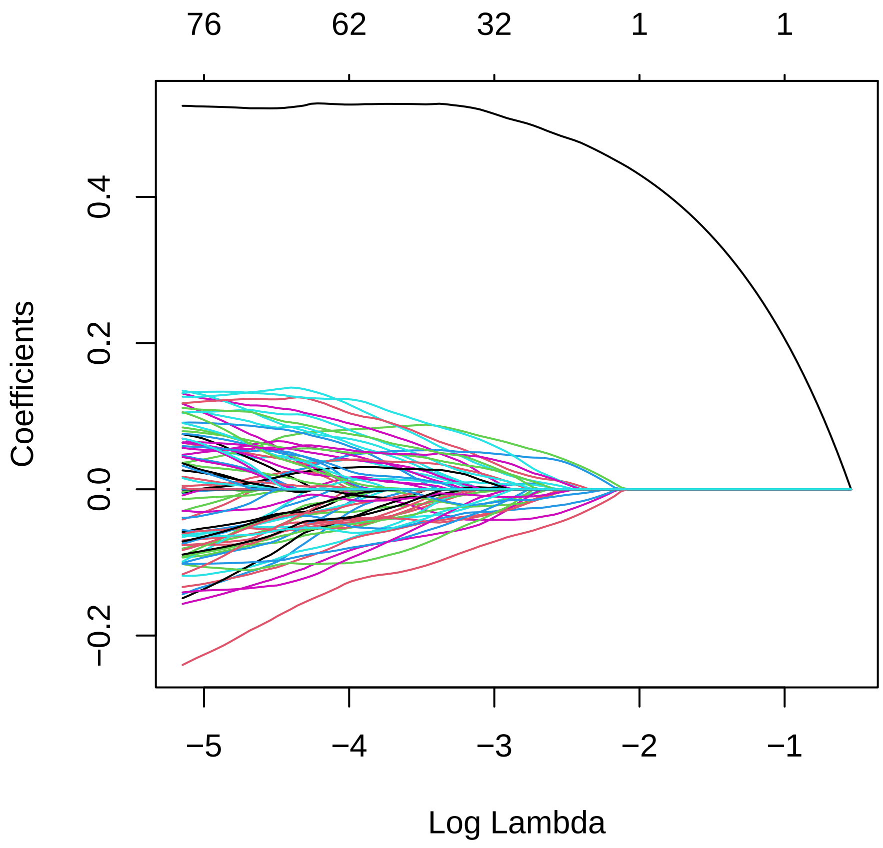
- overfitting:

```
plot(X[,1], Y)
sX1 <- sort(X[,1])
points(sX1, sin(sX1), type="l", col="blue")
points(sX1, fitted(lm(Y ~ X[,1] + I(X[,1]^2) + I(X[,1]^3) +
                        I(X[,1]^4) + I(X[,1]^5) + I(X[,1]^6) +
                        I(X[,1]^7)))[order(X[,1])],
       type="l", col="red")
```



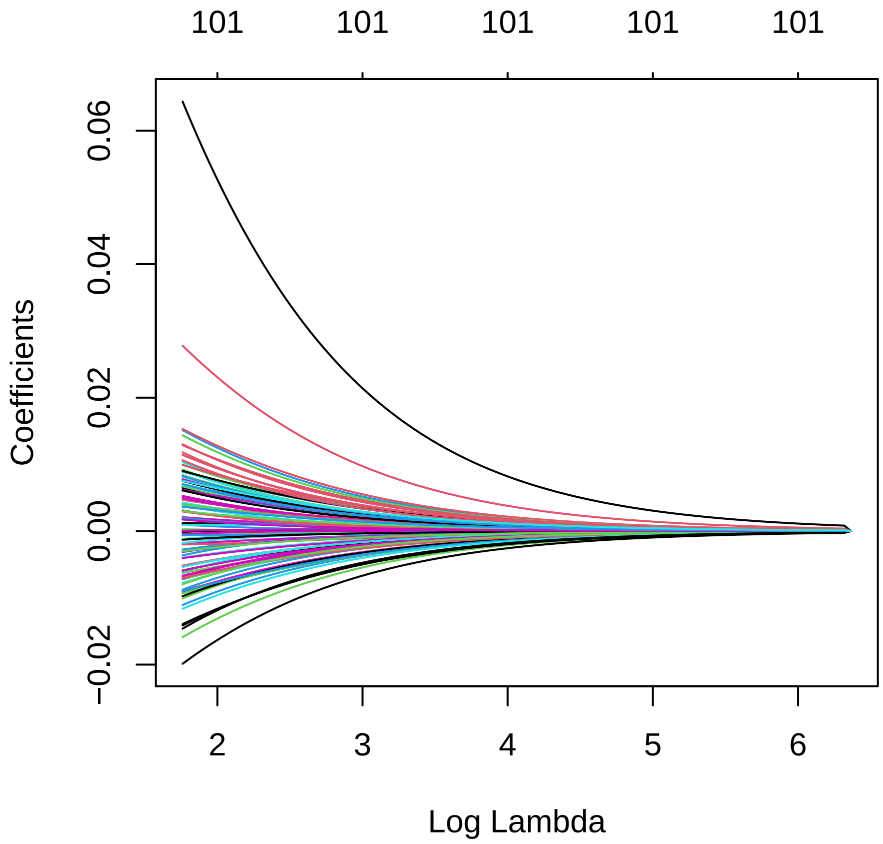
- $L1$ penalty:

```
library(glmnet)
plot(glmnet(X, Y), xvar = "lambda")
```



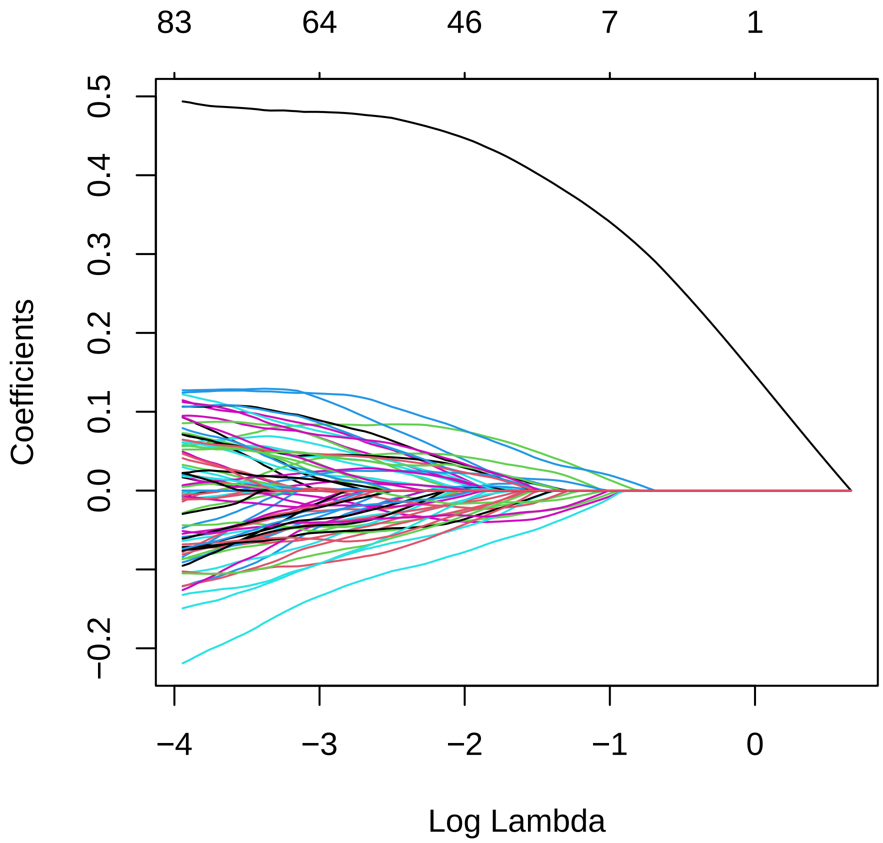
- $L2$ penalty

```
plot(glmnet(X, Y, alpha = 0), xvar = "lambda")
```



- elastic net regularization:

```
plot(glmnet(X, Y, alpha = 0.3), xvar = "lambda")
```



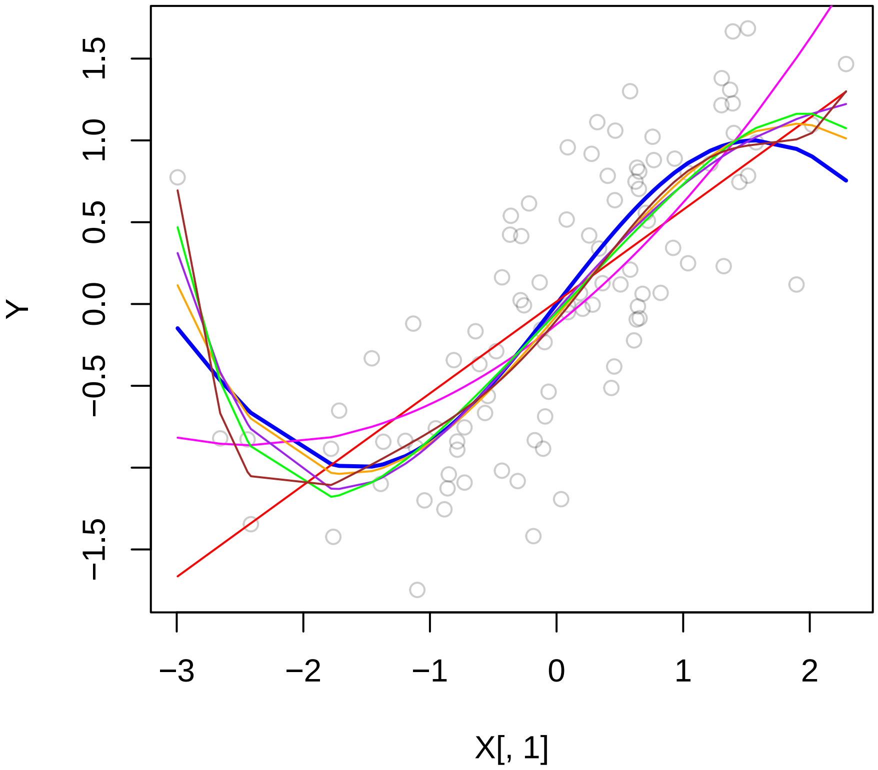
- the underdetermined problem:

```
try(ls_estimator <- solve(crossprod(X), crossprod(X,Y)))

## Error in solve.default(crossprod(X), crossprod(X, Y)) :
##   System ist für den Rechner singulär: reziproke Konditionszahl = 5.84511e-18
```

- the bias-variance trade-off:

```
plot(X[,1], Y, col=rgb(0,0,0,0.2))
sX1 <- sort(X[,1])
points(sX1, sin(sX1), type="l", col="blue", lwd=2)
points(sX1, fitted(lm(Y ~ X[,1]))[order(X[,1])],
       type="l", col="red")
points(sX1, fitted(lm(Y ~ X[,1] + I(X[,1]^2)))[order(X[,1])],
       type="l", col="magenta")
points(sX1, fitted(lm(Y ~ X[,1] + I(X[,1]^2) + I(X[,1]^3)))[order(X[,1])],
       type="l", col="orange")
points(sX1, fitted(lm(Y ~ X[,1] + I(X[,1]^2) + I(X[,1]^3) +
                        I(X[,1]^4)))[order(X[,1])],
       type="l", col="purple")
points(sX1, fitted(lm(Y ~ X[,1] + I(X[,1]^2) + I(X[,1]^3) +
                        I(X[,1]^4) + I(X[,1]^5)))[order(X[,1])],
       type="l", col="green")
points(sX1, fitted(lm(Y ~ X[,1] + I(X[,1]^2) + I(X[,1]^3) +
                        I(X[,1]^4) + I(X[,1]^5) + I(X[,1]^6)))[order(X[,1])],
       type="l", col="brown")
```



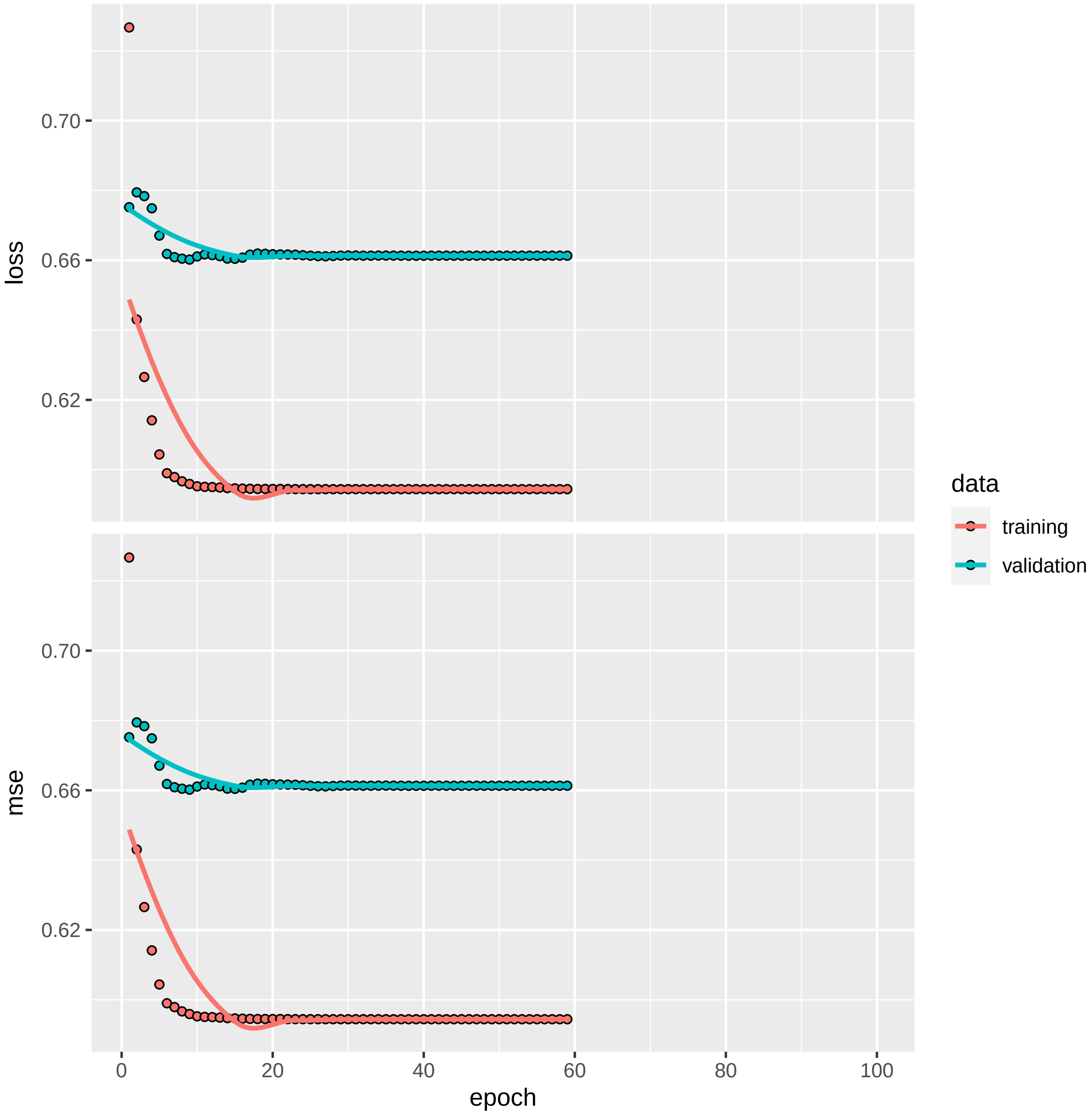
- early stopping using a simple neural network:

```
library(dplyr)
library(keras)

neural_network <- keras_model_sequential()

neural_network %>%
  layer_dense(units = 50, activation = "relu") %>%
  layer_dense(units = 50, activation = "relu") %>%
  layer_dense(units = 1, activation = "relu") %>%
  compile(
    optimizer = "adam",
    loss      = "mse",
    metric = "mse"
  )

history_minibatches <- fit(
  object           = neural_network,
  x                = X,
  y                = Y,
  batch_size       = 24,
  epochs           = 100,
  validation_split = 0.2,
  callbacks = list(callback_early_stopping(patience = 50)),
  verbose = FALSE, # set this to TRUE to get console output
  view_metrics = FALSE # set this to TRUE to get a dynamic graphic output in RStudio
)
plot(history_minibatches)
```


0.70
0.66
0.62
loss
0.70
0.66
0.62
mse
0
20
40
60
80
100
epoch
data
training
validation